

Pre- and Post-Tests for Reusable Learning Objects

H.- Dietrich Wuttke
 TU Ilmenau
 Max-Planck-Ring 14
 98693 Ilmenau
 Telephone +49 3677 692820
 dieter.wuttke@tu-ilmenau.de

Karsten Henke
 TU Ilmenau
 Max-Planck-Ring 14
 98693 Ilmenau
 Telephone +49 3677 691443
 dieter.wuttke@tu-ilmenau.de

ABSTRACT

Because the production of high quality, media supported learning objects is a very cost and time-consuming process, companies are searching for strategies and policies to reuse such valuable content in different scenarios and to get so called reusable learning objects (RLOs). Supporting learning as well as teaching by electronic media is a form of reuse of learning objects, if the same objects can be applied without a change of code. In the paper, we would like to present a Web-based test, examination and assessment system as well as some examples of reusable learning objects and discuss possibilities to adapt learning objects to suit different learning scenarios. The learning objects can be used via Internet but are suitable for both, local and remote examination preparation as well as for an examination within a lab course as an entry test.

Categories and Subject Descriptors

K.3.1 [Computer Uses in Education]: *Computer-assisted instruction, Distance learning.*

General Terms

Algorithms, Management, Measurement, Design, Experimentation, Human Factors, Standardization, Languages.

Keywords

Assessment system, Reusable learning objects.

1. INTRODUCTION

Produce or reuse a needed content? - That is a question that gets more and more weightiness. The efficiency of the production can increase (or decrease) the profit significantly. That's why companies and educational institutions are searching for strategies and policies to reuse such valuable content. They are looking for methods and standards to use content in different scenarios and to get so called reusable learning objects (RLOs). Existing standards describe the reusability from a very technical point of view. They define mainly the exchange of RLOs between Learning Management Systems (LMS). A huge amount of data structure definitions is given for example in the Shareable Courseware Reference Model (SCORM). They are useful for developers of Learning Management Systems and authoring tools but less suitable for authors and teachers.

Authors need a standardized ontology as well as policies to describe the main properties of the learning content. They need categories describing the use and reuse of learning objects from the learner's as well as from the teacher's point of view. Categories as learning out-comes, pre and post competencies, learning style and so on are much more of interest than technical details.

Before producing content, it is necessary to investigate if a similar content exists and, if yes, what effort is needed to adapt the content to the new requirements. Content can only be reused if the necessary effort is lower than the effort needed for a new production.

In this paper, we want to describe properties of RLOs as well as levels of reusability, which can be useful during the design process from the author's point of view. First, we introduce reusable learning objects. We continue with a collection of examples for the reuse of learning objects concerning to different learning scenarios and present a concept for a Web-based and database supported e-Learning Test, Examination and Assessment System. This system is - among other things - suitable for a remote lab entry test. It is developed at the Department for Integrated Hardware and Software Systems at the Technical University of Ilmenau.

2. REUSABLE LEARNING OBJECTS

The following definition, taken from [1], gives a practicable description of a RLO: "A reusable learning object (RLO) is based on a single learning objective, comprising a stand-alone collection of three components:

1. **Content:** a description of the concept, fact, process, principle or procedure to be understood by the learner in order to support the learning objective
2. **Interaction:** something the learner must do to engage with the content to achieve a better understanding
3. **Assessment:** a way in which the learners can apply their understanding and test their mastery of the content".

This definition fits with CISCO®'s RLO-definition, shown in figure 2, as well as with a refined model made by Uskov in [2]. The components of an RLO are called reusable information objects RIOs or reusable information atoms (RIAs).

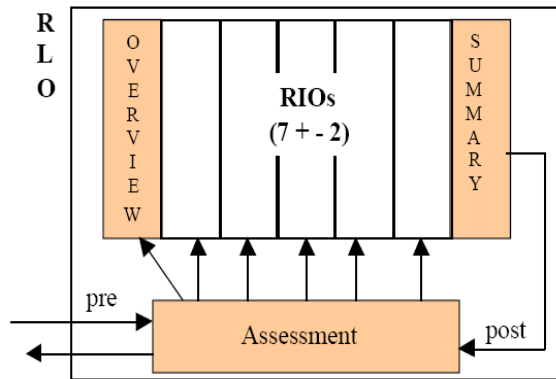


Figure 1: RLO Defined by CISCO® [3].

Additional to this classification we would like to introduce a pre- and post- test for each RLO. The pre-test should give the user an impression if he should pass this learning object or not. It also should test if the learner has the required pre-knowledge to pass the learning object without the use of further materials. The post-test should assess the learning outcomes and give user a feeling about the level of knowledge he/she has reached.

For such kind of pre- and post-tests, we have developed a Web-based and database supported e-Learning Test, Examination and Assessment System, which we will describe in the next section.

3. TEST, EXAMINATION AND ASSESSMENT SYSTEM

3.1 Requirements

During the development of the *Test, Examination and Assessment System* priority was given to platform independence. The system should be usable from any place at any time by different users with different system properties. Beyond that there should be the possibility to integrate this system into Learning Management Systems (LMS). At various universities there have been established diverse LMS. While planning the *Test, Examination and Assessment System* we choose not to implement mechanisms for a special LMS. Therefore a realization as Java Applets was selected as common platform offering the chance to use the system stand-alone without Internet connectivity in addition.

The *Test, Examination and Assessment System* is generally to be used for examinations (e.g. entry test for online labs). Furthermore it should allow exercising learning content anonymously (which means without any login procedure and grading). Tasks and examinations should be managed by database support.

An assessment happens either by displaying the correct results (during *exercise mode*) or by calculating a mark (during *examination mode*).

During the implementation of the *Test, Examination and Assessment System* great importance was attached to the

- reuse of the applied methods and tools for similar problems,
- reuse of the developed software modules,

what finally leads to synergy effects in the production process of different test and assessment tools.

3.2 Architecture

The *Test, Examination and Assessment System* essentially consists of the following components:

- the test system AWiP¹ itself,
- a database supported Collection of Tasks,
- the Task Assembler and
- an Administrator Tool.

The complete system is implemented multilingual. All text messages and labels are stored in an extra file and can be exchanged during run time.

Another feature is the possibility to modify the system for different requirements (e.g. different courses of study, lectures and labs) without any change of code.

The AWiP system provides tasks from a (randomly selectable) text file and performs an evaluation of the student's result. The student can solve his tasks in any sequence and is able to interrupt his work for a certain time.

At present the following types of tasks are implemented:

- Multiple Choice Tasks (see Figure 2)
- Cloze Tasks (see Figure 3).

The concept for the whole system is built in a way that new types of tasks can be implemented in perspective by using simple and common available text editors. It is also possible to place additional graphics inside the text of the tasks.



Figure 2

AWiP Example for Multiple Choice (with Check)

¹ AWiP stands for „Applet zur Wissensüberprüfung im Praktikum“ - in English: “An applet for knowledge testing in laboratory courses”.

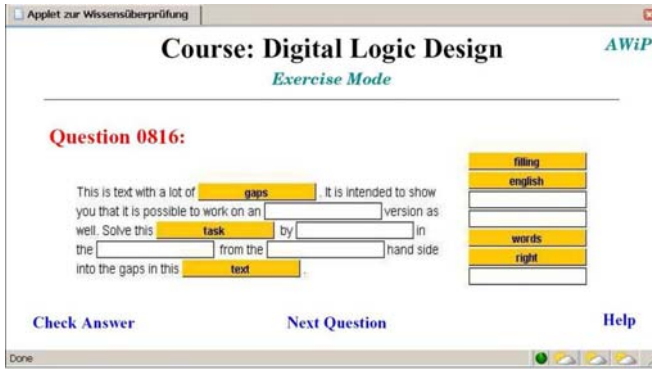


Figure 3

AWiP Example for Cloze TASKS (with Check)

The multiple choice task, shown in Figure 2, is derived from the following source code segment:

```
# Example for a Multiple Choice Task
[choice]
When you have read the text on top of
this page carefully, you can answer
this question easily. Multiple choice
questions here always have
- one right answer.
- more than one right answer.
+ one or more right answers.
- no right answer.
```

and the cloze task in Figure 3 is generated by:

```
# Example for a Cloze Task
[cloze]
This is text with a lot of $$gaps$$ .
It is intended to show you that it is
possible to work on an $$English$$
version as well. Solve this $$task$$
by $$filling$$ in the $$words$$ from
the $right$$ hand side into the gaps
in this $$text$$ .
```

The following working modes are supported at the present time:

Offline Test

Here the student can generate a collection of tasks suited for his requirements by using the Task Assembler (see section 3.3). Later he is able to solve the given tasks and get an assessment of the results without Web support by using only the AWiP test system.

Online Test

In this case the student is given a number of tasks (either randomly or suited for his requirements) by the Task Assembler,

that he can process online and get an assessment for by the AWiP system corresponding to his answers.

Examination

During examination mode a log file will be generated automatically in the background that will record the actual state of the student's work. That's why a login procedure (with name and student ID) is required at the beginning of the examination. After that the student is given his tasks to be processed in a given time in any order. When all tasks are solved (or the time limit is over) the resulting mark will be calculated (via a comparison with an existing list of marks) and displayed. As a proof the student will get a certificate with relevant data (course, name, student ID, mark, date).

While the AWiP system is in examination mode it can only be closed by a tutor (the supervisor in the examination room) using his admin password. During this closing procedure the created log file will be removed. If there is an unexpected break (without admin password) the exact state before the termination will be restored during an AWiP restart.

AWiP is realized as a Java-Applet using PHP scripts for file handling. The core is an interpreter module for the text-messages of the task files.

3.3 Data based Task Collection

All tasks and examinations show a simple syntax and are completely file based. That's why it is possible to process these files by any editor (offline as well). This is an important advantage compared to integrated QTE² editors of Learning Management Systems (LMS) [4]. Required attributes and metadata were determined by performing an analysis of demands and use cases [5].

Every task file of the *Web-based Test, Examination and Assessment System* consists of a simple text file separated in one or more segments. Each segment begins with a type identifier in square brackets. A single task collection can contain any task of different types. At the present time the following types are supported:

[text]	The following text will be displayed without any formatting.
[choice]	The segment will be interpreted as a multiple choice task. Lines starting with ``+'' or ``-'' will be displayed as choice options.
[cloze]	The segment is a cloze task. Terms to be inserted will be enclosed by \$\$.
[image]	The following line gives the path to an image file.

For the administration of all data the data bank MySQL 5.0 was chosen [6].

² Question and Task Editor

3.4 Task Assembler

The *Task Assembler* defines in which manner the tasks - stored in the data base - will be offered. It can be used and configured by students themselves as well as by the lecturer (who is responsible for the course).

The student is given the possibility to get an assembly of tasks

- either randomly or
- *individually* by using metadata

for exercise purposes.

This assembly of tasks (a simple text file) can be processed using the *AWiP* test system directly online or at a later time offline.

The lecturer can assemble the following:

Examination

Through the selection of different metadata (e.g. main topics, grade of difficulty, duration of processing, number of points, coding of the marks, ..., see section 3.6) he can create a task assembly for an examination. He sends this collection to the supervisor, responsible for that examination (tutor) using a secured connection.

Examination Library

A library of examinations is a collection of single examinations for the same topic with the same constraints. That way it is possible to offer different versions of an examination if multiple students have to be examined in one single room.

Written Examination

Tasks for a written examination can be generated and are also selected by metadata. The result in this case is a "print-ready" file.

3.5 Administrator Tool

For an effective handling of the whole *Web-based Test, Examination and Assessment System* the test system *AWiP* itself was enhanced by a Web-based *Administrator Tool* for the preparation of tasks, examinations and metadata. It offers the following features:

- to create and update tasks using a build-in or external editor,
- to import and export tasks,
- to assemble examinations and libraries of examinations,
- to define metadata for tasks, examinations and libraries of examinations.

3.6 Metadata

Besides the above mentioned types of tasks (see section 3.2), metadata will be used for a universal configuration of the "*Web-based Test, Examination and Assessment System*" to allow a later integration into a LMS as well.

To determine the required metadata both the demands of the described system and the structure of other systems based on the LOM³ standard (e.g. GETsoft [7]) were taken into consideration.

The LOM standard does not fit exactly the objectives of the described system. But it offers a suitable template for learning objects - a generic term for enclosed e-Learning units like exercises, simulations, simple text and so on. This template defines a multitude of attributes. Some of them appear to be not practical for the object types (task, examination, library of examination) used in the *Web-based Test, Examination and Assessment System*. The same applies for possible enhancements like the addition of new task types or the database usage as repository for written examinations.

Furthermore the LOM standard offers no hierarchy as it would be necessary for the described system. The terms "*task*", "*examination*" and "*examination library*" define completely different data structures. A common data type "*learning object*", containing all attributes of subtypes and additional properties declared by the LOM standard, would provide a multitude of unneeded metadata.

An inquiry regarding WBT⁴ produced the result, that the used attributes in Web-based examination systems are very similar to each other [5]. Some of them were used here - others were abandoned in order to stay with the essential facts. The following table contains the chosen metadata for the type "*task*". The metadata used for the types "*examination*" and "*examination library*" are listed in [8].

Identifier	Integer value as unique ID of a task
Type	Type of the task (simple text, multiple choice, cloze); further types can be declared without any software changes.
Theme	For a better structuring of the content a subject will be assigned to each task.
Course	A course is defined by its name and its corresponding study courses. One task can be allocated to various courses.
Semester/ Year	To handle different versions of the database material over a longer period.
Modus	Defined modes are (at the moment): exercise only, examination only, exercise together with examination and supplementary tasks during examination.
Difficulty	One out of five levels can be selected.
Points	Suggested number of points
Duration	The guessed processing time (in minutes)
Language	The choice of language.

³ The Learning Objects Metadata (LOM) standard provides a uniform system for attributes of a learning object. A learning object can be any possible form of an e-Learning content, like examinations or simulations [9].

⁴ Web-based Training

Author	Contact Data (e.g. Name, Phone and email)
Keywords	A list of terms for quick search.
Release	Is the task usable for examinations or locked?
Version	The date of last change.
Comment	Some additional comments for administration.
Outcomes	Learning outcomes are used to decide where to branch into the next learning steps, depending on the actual result.
AddOns	Reserved for the implementation of Add-Ons.

4. REUSE OF RIOS FOR ASSESMENT

Concerning to the RLO structure shown in Figure 1 we want to give some examples of learning objects tailored to special purposes of the educational process. Core of the learning objects are applets that we have discussed in former publications (see for instance [10]). These applets follow the concept of “Living Pictures” where the big picture (a summary of a lesson) leads to the highly inter-active user interface of the learning object.

We use them in different learning scenarios:

- During a presentation in a face to face lesson
- As interactive tools for the support of the students self study and own experiments
- As interactive tasks for an assessed problem solving

To use these applets for presentation purposes in a lecture they are designed in a way that a presenter is able to ex-plain their functionality with some mouse clicks. Default produced examples support this demonstration mode and free the presenter from boring input strings.

Figure 4 shows an example of a living picture which can be used to teach the principle of programmable logic devices.

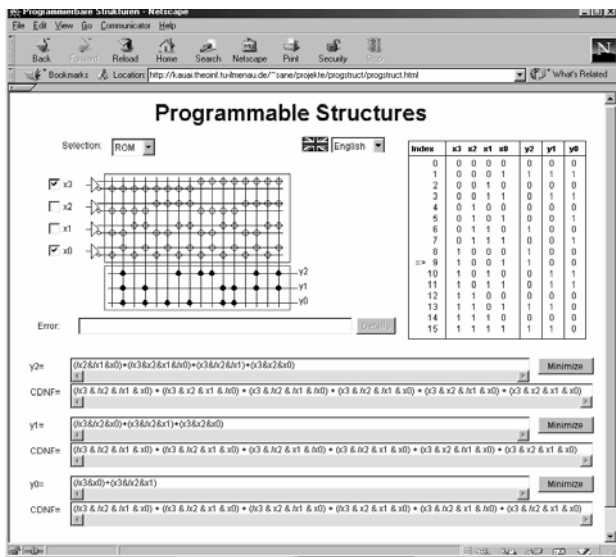


Figure 4 Living Picture “Programmable Structures”

It simultaneously displays three different views of a programmed function: A truth table (at the right hand side of the screen shot), a set of Boolean equations in the upper part of the screenshot and the structure of the programmable device in the upper left part of the living picture. For more details see [10].The same applet can be used as a tool for own experiments, if students repeat the learned subjects at home. In this use case, the applets support an exploration-oriented approach to the learning subject. Student can create their own examples or repeat the examples they have seen during the lesson.

Based on this applet we have derived a number of interactive tasks the students can use for an initial, intermediate or final assessment of their knowledge. These kind of interactive tasks give the students a greater variety of answers as the multiple choice and cloze tasks discussed in section 3 and thus deeper understanding of the learning subject.

To derive such interactive tasks the simulation component of the living picture was modified by a programmer. This kind of reuse requires special knowledge about the program code as well as appropriate programming skills and tools. The applet was modified to write the simulation results in a separate data structure instead of displaying results of experiments immediately. In this way, most of the code of the applet could be reused. Only components for task generation and assessment have to be added.

In these interactive tasks, we present a function described in one of the three above-mentioned views of a programmable structure (truth table, Boolean equation and the structure of the programmable device). The students’ task is to find an equivalent description of the function in one of the other two views.

After starting the exercise, an example is randomly generated where a function is given in one of the three forms. Task for the student is to determine the same function in another view.

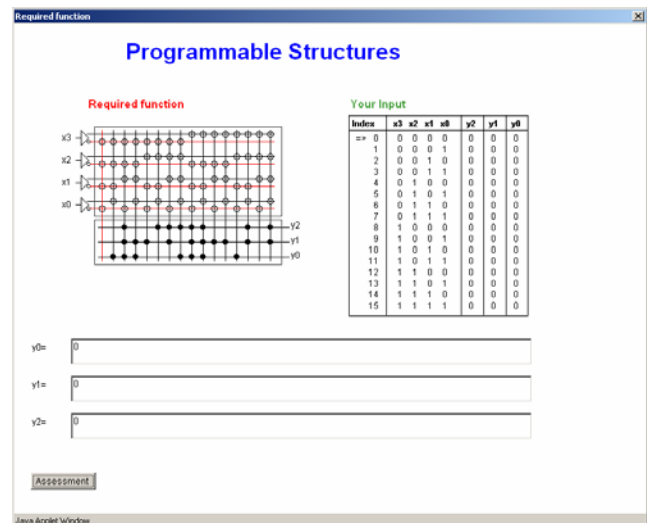


Figure 5 Screenshot for task solving

Figure 5 gives an example where the function is generated in the form of a programmed ROM structure (“Required function”). Thus, the students’ task is to analyze the programmed structure and to find out which values in the truth table have to be set to the

value “1”. For this task, the truth table is clickable and the values of the output functions $y_2\dots y_0$ can be changed by simple mouse clicks (“Your Input”).

If the student has found a solution, he/she can get an assessment of the solution by clicking the appropriate but-ton. The program counts the faulty output values and gives a feedback, showing the differences between the required function and the students’ solution in the required view.

5. CONCLUSION

The two kinds of assessment possibilities described in section 3 and 4 differ in the variety of possible answers as well as in the effort to produce new tasks. Inside the domain “Programmable Structures” the method described in section 4 is very comfortable because a very big amount of tasks (2^{16} possible functions combined with 3 output variables and 3 different views) can be generated automatically. To generate such an amount of multiple-choice questions in the way described in section 3 all variants have to be written in extra text files, a very boring task! Nevertheless, if we leave the domain of “Programmable Structures” another “living Picture” has to be varied in the way described in section 4 by a skilled programmer. Thus booth way have their own advantages and disadvantages and at least the available capacities for the development of reusable learning objects decide, which way can be gone.

6. ACKNOWLEDGMENTS

This work is supported - as a part the project “MARE - Joint Master Degree Programme Remote Engineering” - by the European Commission within the program “ERASMUS - Joint Development of Study Programmes at intermediate and advanced level”, Grant No 29298-IC-1-2004-1-AT-ERASMUS-EUC-1 as well as by the Thuringian Ministry for Science, Research and Art (TMWFK/ TKM).

The authors would also like to thank Jana Quasebarth, Gert Dietrich and Jan Zwanziger for programming the *AWiP* tool as well as Norman Wagner for the development of the corresponding administrator tool. Furthermore, we would like to thank Sven Hellbach and Andreas Degenhardt for programming the applets presented in section 4.

7. REFERENCES

- [1] Leeder, D., Davies, T., & Hall. A.: *Reusable learning objects for medical education: evolving a multi-institutional collaboration*, Retrieved January 23, 2005, from <http://www.ucel.ac.uk/documents/docs/068.pdf>.
- [2] Uskov, V. Design, development and teaching of innovative Web-based introductory "Computer Information Systems" course", *Frontiers in Education, 2002. FIE 2002. 32nd Annual Conference* Volume 3, 6-9 Nov. 2002 Page(s):S1E-13 - S1E-18.
- [3] CISCO, *Reusable Learning Object Strategy - Definition, Creation Process, and Guidelines for Building*, http://www.reusablelearning.org/Docs/Cisco_rlo_roi_v3-1.pdf
- [4] Gouli, E.; Kornilakis, H.; Papanikolaou, K.A., Grigoriadou, M. Adaptive Assessment Improving Interaction in an Educational Hypermedia System, Proceedings of the *Human Computers Interaction 2001*, Panhellenic Conference with International Participation, Patras, Grece
- [5] Ludwig, N. *Learning Environments for Remote Labs - Concept und Realization*, Master’s Thesis, Technical University of Ilmenau, 2005
- [6] MySQL: MySQL website - <http://www.mysql.com>
- [7] Neundorf, V.; Yakimchuk, V. *GETsoft - Multimediale Lernumgebung für Grundlagen der Elektrotechnik*, <http://get-16.e.technik.tu-ilmenau.de/getsoft/index.php>
- [8] Wagner, N.: *Computer Aided Test*, Study Thesis, Technical University of Ilmenau, 2006
- [9] Learning Technology Standards Committee: *WG12: Learning Object Metadata*, <http://ltsc.ieee.org/wg12>
- [10] Wuttke, H.-D., Vangelov, T.: Web-Based Training on the Field of System On Chip Design, *11th International Conference Mixed Design of Integrated Circuits and Systems MIXDES 2004*, Szczecin, Poland, June 24-26th, ISBN 83-919289-7-7, pp. 673-677.